PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04L 9/00	A1	(11) International Publication Number: WO 00/49764
		(43) International Publication Date: 24 August 2000 (24.08.00)

(21) International Application Number: PCT/US00/03960

(22) International Filing Date: 16 February 2000 (16.02.00)

(30) Priority Data:
09/250,935 18 February 1999 (18.02.99) US

(71) Applicant: SUN MICROSYSTEMS, INC. [US/US]; M/S PALI-521, 901 San Antonio, Palo Alto, CA 94303 (US).

(72) Inventors: PERLMAN, Radia, Joy; 10 Huckleberry Lane, Acton, MA 01720 (US). HANNA, Stephen, R.; 3 Beverly Road, Bedford, MA 01730 (US).

(74) Agents: SHEEHAN, Patricia, A. et al.; Cesari and McKenna, LLP, 30 Rowes Wharf, Boston, MA 02110 (US).

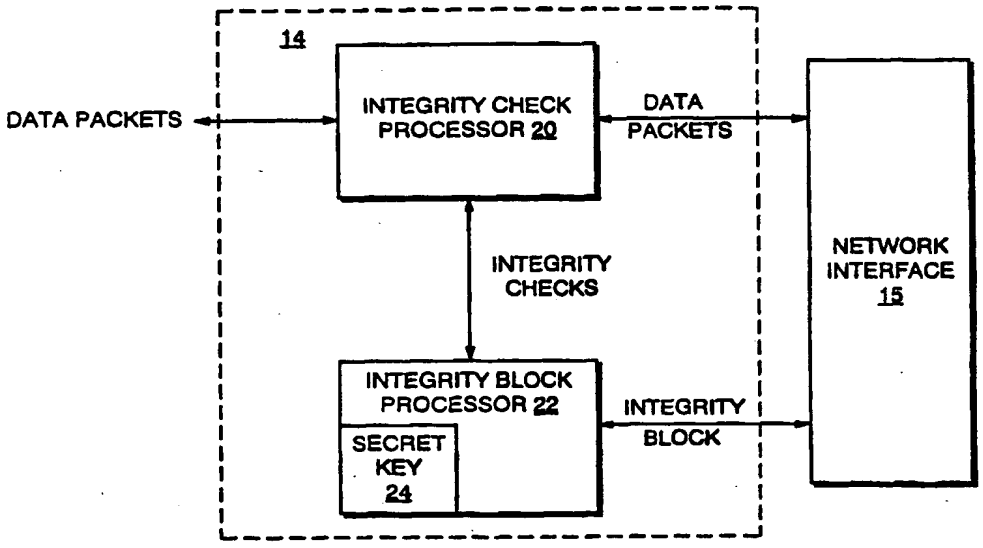
(81) Designated States: AE, AL, AU, BA, BB, BG, BR, CA, CN, CR, CU, CZ, DM, EE, GD, GE, HR, HU, ID, IL, IN, IS, JP, KP, KR, LC, LK, LR, LT, LV, MA, MG, MK, MN, MX, NO, NZ, PL, RO, SG, SI, SK, TR, TT, UA, UZ, VN, YU, ZA, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: DATA AUTHENTICATION SYSTEM EMPLOYING ENCRYPTED INTEGRITY BLOCKS

(57) Abstract

A data authentication system (14) that at the sender produces for a plurality of data packets a plurality of "integrity checks" by selecting a number of bytes from a given packet and either using the bytes directly as an associated integrity check or encoding the bytes to produce an integrity check. The system then selects corresponding bytes or bytes that are offset from the corresponding bytes from a next packet and produces a next associated integrity check, and so forth. The system encrypts the integrity checks associated with the plurality of data packets using, for example, a shared secret key (24), and produces an integrity block. The system then sends the encrypted integrity block and the data packets to the intended recipients. A recipient decrypts the integrity block using the shared secret key and reproduces the integrity checks. It then uses the integrity checks to authenticate the associated data packets. The recipient thus performs a single decryption operation and a plurality of relatively fast integrity check operations to authenticate a plurality of data packets. The sender may also include in a transmission one or more extraneous, or "chaff", data packets, which are data packets that intentionally fail the associated integrity checks (20). The sender may, for example, include in a transmission multiple sets of packets with the same sequence numbers. The recipient readily determines which of the packets with the same sequence numbers are valid using the appropriate integrity check.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

DATA AUTHENTICATION SYSTEM EMPLOYING ENCRYPTED INTEGRITY BLOCKS

FIELD OF INVENTION

The invention relates generally to data communications networks and, in particular, to systems for authentication of data transferred over the networks.

BACKGROUND OF THE INVENTION

When a sender transfers data over a network, an interloper may intercept and alter the data and then transfer the altered data to the intended recipients. The recipients, who presume the data are valid because they appear to come from a trusted sender, may then use the altered data directly, or introduce errors into associated data processing systems. To ensure that the received data were not altered enroute by an interloper, the network may include a data authentication system that essentially encodes the data at the sender and decodes the data at the recipient to detect changes in the data.

One known data authentication process involves including a digital signature in a data packet. The digital signature is produced by first encoding the packet data bytes to produce a cryptographic hash and then, typically, encrypting the hash using the sender's private key. A recipient uses the sender's public key to decrypt the digital signature and reproduce the hash. It then encodes the received data using the same cryptographic hash function and compares the result with the decrypted hash. If the two hashes match, the data is considered authentic, that is, the received data is considered to be the same data that was sent by the sender who holds the private key. This authentication process works well, but it is both computation intensive and time consuming at the recipient end.

A slightly less computation intensive authentication process that may be used by senders and recipients that share a secret key is commonly referred to as a message integrity code, or "MIC," process. The MIC process produces an integrity code by concatenating the shared secret key with the data and then encoding the data and the key using a cryptographic hash function. The result, which is the integrity code, is then sent along with the data to a recipient who shares the secret key. The recipient similarly concatenates the shared secret key with the received data and encodes the data and the key using the hash function. If the result matches the received integrity code, the data is considered authentic. The MIC authentication process works well and is relatively reliable, assuming the holders of the shared key are trusted. However, this process still requires producing a hash based on all of the data bytes in the data packet, and is thus still relatively time consuming.

SUMMARY OF THE INVENTION

The invention is a data authentication system that at the sender produces for a plurality of data packets a plurality of "integrity checks" that it then encrypts with a shared secret key and sends as an "integrity block." A recipient decrypts the integrity block using the shared secret key and reproduces the integrity checks. It then uses the integrity checks to authenticate the associated data packets. As discussed below, the authentication system uses a relatively weak, and thus, fast and uncomplicated, integrity check process because the integrity checks are encrypted, and therefore, unknown to an interloper. The recipient performs a single decryption operation and a plurality of fast integrity check operations to authenticate a plurality of data packets. Accordingly, the authentication system operations are less time consuming and less complex than those of known prior authentication systems.

More specifically, the authentication system selects a number of bytes from a given packet and either uses the bytes directly as an associated integrity check or encodes the bytes to produce the integrity check. The system then selects corresponding bytes or bytes that are offset from the corresponding bytes from a next packet and produces a next associated integrity check, and so forth. The system

encrypts the integrity checks associated with the plurality of data packets using a shared secret key, and produces the integrity block. The system then sends the encrypted integrity block and the data packets to the intended recipients.

A recipient uses the shared secret key to decrypt the integrity block and reproduce the integrity checks. It then associates the integrity checks with the data packets and, for each data packet, selects the appropriate data bytes and authenticates the data using the appropriate integrity check.

If an interloper intercepts the data packets and the integrity block, the interloper cannot readily interpret the integrity block because of the encryption. Accordingly, the interloper cannot readily determine, for example, which bytes are selected for authentication or how to alter the data and still pass the integrity checks without knowing the secret key. The authentication system thus inhibits the actions of the interloper even if only a relatively small number of bytes from a given data packet are used to produce the associated integrity check.

As discussed in more detail below, the integrity block may include information that is used by the recipient to determine, for a given data packet, which of the data bytes were used to produce the associated integrity check. As necessary, the integrity block includes offset values, interval values and so forth. In addition, the integrity block includes information, such as packet sequence numbers, that associate the integrity checks with the appropriate data packets, if the integrity checks in the block are not in the same order as the data packets, or the integrity block is associated with a group of previously sent data packets, or data packets yet to be sent, and so forth. All of the information in the integrity block, including the offset or interval values and the sequence numbers, as appropriate, is preferably encrypted and therefore unknown to the interloper.

As a further safeguard against an interloper, the authentication system may include in the integrity block a digital signature that is prepared using the sender's private key. The recipient can then test that the integrity block has not been altered by an interloper that knows the shared secret key. Further, the sender may include digital

signatures in the data packets so that the recipient can more robustly test selected data packets.

Alternatively, or in addition, the sender may include in a transmission one or more extraneous, or "chaff," data packets, which are data packets that intentionally fail the associated integrity checks. The sender may, for example, include in a transmission multiple sets of packets with the same sequence numbers. The recipient readily determines which of the packets with the same sequence numbers are valid using the appropriate integrity check. However, an interloper who cannot decipher the integrity block cannot as easily determine which of the packets are valid, and thus, cannot determine which packets to alter and/or how to alter these packets without detection.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention description below refers to the accompanying drawings, of which:

Fig. 1 is a functional block diagram of a network constructed in accordance with the invention;

Fig. 2 is a more detailed block diagram of an authentication system that is included in the network of Fig. 1; and

Fig. 3 is a functional block diagram of an alternative authentication system.

DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

Referring to Fig. 1, a communications system 10 includes a plurality of end stations 12 that act as either senders or recipients or both. A given sending end station 12 sends data, in the form of data packets, through a communications network 16 to one or more selected receiving end stations 12 that may, for example, be connected to data processing systems 18.

Each end station 12 includes an authentication system 14. When an end station is sending data, the authentication system 14 included therein uses selected information and data bytes from one or more data packets to produce a corresponding set of one or more integrity checks. The authentication system next encrypts the integrity checks in

a known manner in accordance with a shared secret key, and produces an "integrity block." The end station 12 then sends the data packets and the integrity block over the communications network 16 to one or more recipient end stations 12 through a network interface 15 (Fig. 2) in a known manner. The integrity block may be included in a data packet or it may be sent as a separate packet, as appropriate. The operations to produce the integrity checks and the integrity block are discussed below with reference to Figure 2. Hereinafter, we refer to the end stations 12 as senders and recipients based on their operations in the transmission under discussion.

Referring now to Fig. 2, the authentication system 14 includes an integrity check processor 20 that, during send operations, produces the integrity checks. During receive operations the integrity check processor 20 uses the integrity checks to determine if the received data packets are authentic, as discussed below. An integrity block processor 22, which is also included in the authentication system 14, encrypts the integrity checks and produces the integrity block that is sent over the network to the intended recipients. The integrity block processor 22 also decrypts a received integrity block, to reproduce the integrity checks that are then used by the integrity check processor 20 to authenticate the received data packets.

We discuss below the basic authentication process performed by the authentication system 14 and thereafter several variations of that process. The basic authentication process uses shared secret keys. However, as discussed below, the sender and recipient may also use a public/private key combination for further robustness.

The integrity check processor 20 selects from a first data packet "m" data bytes, with $1 \leq m \leq d$ and where "d" is the number of bytes in the data packet, and manipulates the bytes to produce an associated integrity check. The processor 20 in the example manipulates the selected bytes by concatenating them to produce the integrity check. The processor then supplies the integrity check to the integrity block processor 22, which retains the bytes for encryption. The integrity check processor 20 similarly selects m corresponding bytes from the next "p"-1 data packets, with $p \geq 1$, and supplies them to the integrity block processor 22 as the associated integrity checks.

When all p integrity checks are supplied to the integrity block processor 22, the processor encrypts them in a known manner using the shared secret key that is held in a register 24. The result of the encryption is an integrity block that is transferred over the communications network 16 to the intended recipients of the p data packets.

The integrity block processor 22 at the recipient readily decrypts the received integrity block in a known manner using the shared secret key, and reproduces the integrity checks for the p data packets. The integrity check processor 20 at the recipient then authenticates each of the p received data packets by comparing the bytes in the associated integrity check with the corresponding bytes from the appropriate data packet. If the bytes match, the integrity check processor determines that the given data packet is authentic. The authentication system 14 thus authenticates p data packets by performing a single decryption operation and p m-byte compare operations. This is in contrast with known prior systems that authenticate p data packets by performing p separate authentication operations, each of which typically involves manipulating all of the data bytes in the packet. Accordingly, the system 14 authenticates the data packets faster than the known prior systems. Further, the encoding of the integrity block allows the system to use a less robust, and thus, less complex process to produce the respective individual integrity checks. Accordingly, the system 14 may include less complex circuitry than prior known systems.

As discussed above, the authentication system may produce an encrypted integrity block for a single data packet. If the integrity block is included within the data packet, there will be no delay in authenticating the packet. Otherwise, the system must delay authentication of the packet until receipt of the associated integrity block.

If an interloper intercepts the data packets and the integrity block, it will not be able to decipher the encrypted integrity block, and thus, will not be able to determine which of the data bytes the recipient will use to test the authenticity of a given data packet. Accordingly, the interloper cannot readily determine from the integrity block how to alter a data packet and still satisfy the associated integrity check, even though the integrity check process is relatively weak. The integrity block, even if it relates

only to a single data packet, is thus a relatively robust authentication tool because of the encryption of the individual integrity checks.

Further, the authentication process is not particularly complex or time consuming at the recipient end because the decryption operation is performed once for p data packets, with the decryption operation involving relatively few bytes. Also, the p integrity check operations each involve a relatively small number of data bytes from each packet, and are thus easily and quickly performed. This is in contrast to the known prior authentication techniques, which manipulate all of the data bytes of each of the p data packets to produce the integrity checks and must thus manipulate all of the data bytes of each of the p data packets to test the authenticity of the p packets. Accordingly, the decryption circuitry of the system 14 is less complex than that of the prior system.

We discuss below variations of the basic authentication process. Each variation adds more robustness to the process and, consequently, requires more time and adds complexity to the process at the recipient end.

As discussed above, the integrity check processor 20 at the sender selects m data bytes from each data packet for the integrity checks. If corresponding bytes from each packet are used, an interloper who learns which particular bytes are selected can then alter one or more of the other bytes in a packet without detection. Similarly, if the interloper learns that only a relatively small number of bytes at, for example, the start of the packet, are selected for the integrity check, the interloper can then alter the data bytes at the end of the packet without detection. Accordingly, the authentication system may instead select the data bytes at random from a first data packet and select from subsequent data packets bytes that are offset from those selected from the first packet. The system then includes in the encrypted integrity block information such as offset and interval values, and so forth, to inform the recipient which bytes to select for integrity checking from each of the received data packets.

Alternatively, the system may select data bytes at random from each of the data packets, and include in the integrity block a random number seed value or other

information that the recipient needs to reproduce the random byte selections for the various data packets.

The interloper presumably will not be able to interpret the information in the integrity block, and thus, will not be able to determine which bytes of a given data packet will be used to test for authenticity. The interloper is thus inhibited in its efforts to alter the data without detection, even if it knows that only a small number of data bytes are selected from each data packet to test for authenticity.

As a further safeguard, the integrity check processor may encode the selected bytes, that is, it may combine all or some of the selected bytes by, for example, XORing one or more of the bytes, or performing ones complement addition with some or all of the bytes, selectively multiplying or adding the bytes, or performing similar logic functions, and use the results as the integrity check. For even greater protection, the integrity check processor 20 may instead or in addition use a cryptographic hash function to encode the selected bytes or some combination of them.

If the selected data bytes are encoded, the integrity check processor at the recipient similarly encodes the corresponding bytes from a received data packet and then compares the results with the associated reproduced integrity check. This encoding at the recipient end is not particularly time consuming or complex, however, because relatively few data bytes are involved.

Before encrypting the integrity checks, the integrity block processor may arrange them, encoded or not, in a different order than the associated data packets. At the same time, the system encrypts into the integrity block information that the recipient can use to reassign the integrity checks to the appropriate data packets. If, for example, the data packets include sequence numbers, the integrity block processor may include in the integrity block a list of sequence numbers that associates the integrity checks with the data packets. Alternatively, each integrity check may include the sequence number of the associated data packet. The system may also include in a transmission an integrity block that contains the integrity checks for, for example, a next set of data packets or a previously sent set of data packets. The recipient then uses

the information, such as the sequence numbers, included in the integrity block to determine how to assign the integrity checks reproduced from the decrypted integrity block to the appropriate data packets.

An interloper will presumably not be able to associate the out-of-order integrity checks with the appropriate data packets because of the encryption. Accordingly, the interloper will not be able to determine how to alter the various data packets without detection.

The system may encrypt into the integrity block other information that a recipient requires to properly use the integrity checks. For example, the system may include in the integrity block executable code that implements a new integrity check process. The recipient then decrypts the integrity block to reproduce the individual integrity checks and the associated code, and runs the code to authenticate the data packets using the reproduced integrity checks. The system thus need not send the executable code separately to each recipient.

An interloper, even one who learns much about the old integrity check process, will not be able to decipher the code for the new process and will, therefore, not be able to determine how to alter the data bytes to avoid detection by the new integrity check process. For greater robustness, the sender may sign the code in a known manner with its private key in a known manner, so that the recipient can be sure that an interloper has not altered the code.

Alternatively, the authentication system may include in the integrity block a digital signature that is produced in a known manner using the sender's private key. Before decrypting the integrity block, the recipient authenticates the integrity block using the sender's public key to ensure that the block has not been altered or forged.

For even further robustness, a sender may include in each data packet a digital signature that is encrypted with the sender's private key. A recipient can then decrypt a digital signature with the sender's public key, to double-check the authenticity of a packet that has survived the weaker integrity check process. The recipient can thus use the more time consuming and complex, but stronger, authentication tool to spot check

selected packets. With the spot check, the recipient can also detect data that has been forged by an interloper who has access to the shared secret key.

As discussed, the integrity blocks may be encrypted in a known manner, which typically involves providing to the recipients separate initialization vectors for each block. The vectors are used in a known manner to set registers in the encryption hardware to a desired value before the hardware is used to encrypt, in this case, the integrity checks. To optimize the decryption process, the authentication system may instead use the system or method described in a co-pending United States Patent Application, Serial No. 08/988,520 entitled Secure Communications Channel System and Method, which allows a recipient to derive the initialization vector directly from the received integrity block. The authentication system thus avoids having to send the vectors separately.

Generally, the interloper can not readily access the data packets. If this is not the case, the interloper may be able to capture one or more of them and produce packets that are only slightly altered, and may thus pass the integrity checks. To further inhibit the interloper, an alternative authentication system 14 depicted in Fig. 3 includes, in addition to the processors discussed above, a chaff processor 30 that introduces into the transmission one or more extraneous, or "chaff," data packets that intentionally fail the integrity checks. The chaff data packets have the same sequence numbers as the valid data packets, and an interloper without access to the information contained in the integrity block can not readily determine which of the same-numbered packets is the valid data packet. Accordingly, the interloper can not readily determine which packet to alter to produce a packet that may pass the integrity checks. The recipient, however, readily determines which of the packets are valid data packets, based on the integrity checks, and ignores the rest.

The chaff packets may be used with any of the authentication process variations discussed above. Namely, the chaff packets may be included in a transmission whether or not the integrity checks are concatenations of the selected data bytes or the results of the encoding of the selected data bytes, whether or not the integrity checks are assembled out of order, and so forth.

The foregoing description has been limited to a specific embodiment of this invention. It will be apparent, however, that other processors and various forms of memory, including computer readable media, may be used for storing and executing program instructions pertaining to the techniques described herein. It will be further apparent that variations and modifications may be made to the invention, such as combining one or more of the processors listed separately into a single processor, including in the system stations that produce the encrypted integrity blocks but do not decrypt the integrity blocks and stations that decrypt the integrity blocks but to not encrypt them, and so forth, with the attainment of some or all of its advantages. Therefore, it is the object of the appended claims to cover all such variations and modifications as come within the true spirit and scope of the invention.

What is claimed is:

CLAIMS

1. A data authentication system comprising:

A. an integrity check processor that manipulates m selected data bytes from each of one or more data packets to produce one or more integrity checks that correspond to the one or more data packets; and

B. an integrity block processor that encrypts the one or more integrity checks produced by the integrity check processor and produces an integrity block that is used to authenticate the data packets.

2. The data authentication system of claim 1 wherein the integrity block processor encrypts the integrity checks in accordance with a secret key that is shared by intended recipients of the data packets.

3. The data authentication system of claim 1 wherein the integrity check processor selects the m data bytes at random from a first data packet, and for any remaining data packets selects data bytes that are offset from the data bytes selected from the first data packet.

4. The data authentication system of claim 1 wherein the integrity block processor encrypts into the integrity block information that identifies the data bytes selected from each of the data packets.

5. The data authentication system of claim 4 wherein the information includes data byte interval and offset values.

6. The data authentication system of claim 1 wherein the integrity check processor includes in the integrity checks one or more sequence numbers that are associated with the data packets.

7. The data authentication system of claim 1 wherein the integrity block processor assembles the plurality of integrity checks in an order that differs from the order of the data packets and encrypts into the integrity block information that associates the integrity checks with the appropriate data packets.

8. The data authentication system of claim 7 wherein the integrity block processor encrypts into the integrity block a list of sequence numbers that corresponds to the order of the integrity checks within the integrity block.

9. The data authentication system of claim 1 wherein the integrity check processor produces digital signatures for one or more of the data packets and includes the digital signatures in the respective data packets.

10. The data authentication system of claim 1 wherein the integrity block processor produces a digital signature for the integrity block and includes the digital signature in the integrity block.

11. The data authentication system of claim 1 wherein the integrity check processor concatenates the selected data bytes from a given data packet to produce the associated integrity check.

12. The data authentication system of claim 1 wherein the integrity check processor encodes the selected bytes from a given data packet to produce the associated integrity check.

13. The data authentication system of claim 1 further including a chaff processor for producing for transmission extraneous packets that are associated with and do not pass one or more of the integrity checks, the chaff processor including the extraneous packets in a transmission that includes the data packets.

14. The data authentication system of claim 1 wherein the integrity block processor encrypts into the integrity block executable code that performs an integrity check process.

15. The data authentication system of claim 14 wherein the integrity block processor signs the executable code with a digital signature.

16. A communications network comprising:

A. one or more sending stations for sending data packets;

B. one or more recipient stations for receiving the data packets sent by the sending stations; and

C. an authentication system that includes

i. an integrity block processor for encrypting one or more integrity checks that are associated with one or more data packets to produce an integrity block and including the integrity block in a transmission to the recipient stations, and

ii. authentication means for decrypting a received integrity block to reproduce the one or more integrity checks and using the reproduced integrity checks to determine if data in the associated one or more data packets have been altered.

17. The communications network of claim 16 wherein the authentication system includes an integrity check processor that manipulates one or more selected data bytes from a given data packet to produce the corresponding integrity check.

18. The communications network of claim 17 wherein the authentication means uses the one or more integrity checks and the selected data bytes from the one or more data packets to determine if the data packets have been altered.

19. The communications network of claim 16 wherein the integrity block processor is included in each of the one or more sending stations and the authentication means is included in each of the one or more recipient stations.

20. The communications network of claim 16 wherein the integrity block processor encrypts the integrity checks and the authentication means decrypts the integrity blocks in accordance with one or more secret keys that are shared by the sending stations and the intended recipient stations.

21. The communications network of claim 16 wherein the integrity block processor selects one or more data bytes at random from a first data packet and selects from the remaining data packets data bytes that are offset from the data bytes selected from the first data packet.

22. The communications network of claim 16 wherein the integrity block processor encrypts into an integrity block information that identifies data bytes selected from each of the one or more data packets by the integrity block processor.

23. The communications network of claim 22 wherein the information includes data byte interval and offset values.

24. The communications network of claim 16 wherein the integrity block processor further includes in the integrity block sequence numbers that correspond to the associated data packets.

25. The communications network of claim 16 wherein the integrity block processor assembles the integrity checks in an order that differs from the order of the associated data packets and encrypts into the integrity block information that associates the integrity checks with the appropriate data packets.

26. The communications network of claim 25 wherein the integrity block processor further encrypts into the integrity block a list of data packet sequence numbers that corresponds to the order of the integrity checks within the integrity block.

27. The communications system of claim 17 wherein the integrity check processor further produces a digital signature for each data packet and includes the digital signature in the data packet.

28. The communications system of claim 17 wherein the integrity check processor concatenates selected data bytes from a given data packet to produce the associated integrity check.

29. The communications system of claim 17 wherein the integrity check processor encodes selected bytes from a given data packet to produce the associated integrity check.

30. The communications system of claim 16 further including a chaff processor that produces for transmission one or more extraneous packets that are associated with and do not pass one or more of the integrity checks, the chaff processor including the extraneous packets in a transmission with the associated data packets.

31. The communications system of claim 16 wherein the integrity block processor further includes in the integrity block executable code that performs a integrity check process.

32. The communications system of claim 31 wherein the integrity block processor includes in an integrity block a digital signature that corresponds to the executable code.

33. A method of authenticating data that is sent in data packets, the method including the steps of:

A. manipulating selected data bytes from a first data packet to produce an integrity check;

B. encrypting the integrity check to produce an integrity block;

C. sending the integrity block to intended recipients.

34. The method of claim 33 further including the steps of:

D. decrypting a received integrity block to reproduce the integrity check;

E. using the reproduced integrity check to determine if the first data packet is authentic.

35. The method of claim 34 further including the steps of

i. manipulating data bytes from additional data packets to produce additional integrity checks;

ii. encrypting the additional integrity checks into the integrity block;

iii. decrypting the received integrity block to reproduce the additional integrity checks; and

iv. using the reproduced additional integrity checks to determine if respective additional data packets are authentic.

36. The method of claim 35 further including in the step of encrypting the integrity checks, performing the encryption in accordance with a secret key that is available to the recipients.

37. The method of claim 36 further including in the step of decrypting the integrity block, decrypting the block in accordance with the secret key.

38 The method of claim 35 wherein the step of manipulating data bytes selects the data bytes at random from the first data packet and selects from the additional data packets data bytes that are offset from the data bytes selected from the first data packet.

39. The method of claim 35 wherein the step of encrypting the integrity checks further includes encrypting into the integrity block information that identifies the data bytes selected from the data packets.

40. The method of claim 35 further including in the step of encrypting the integrity checks the step of encrypting into the integrity block data byte interval and offset values.

41. The method of claim 35 wherein the step of manipulating the data bytes to produce the integrity checks further includes the step of including in the integrity checks sequence numbers that correspond to the associated data packets.

42. The method of claim 35 wherein the step of encrypting the integrity checks includes assembling the integrity checks in an order that differs from the order of the associated data packets.

43. The method of claim 42 wherein the encrypting step further includes the step of encrypting into the integrity block a list of sequence numbers that corresponds to the order of the integrity checks.

44. The method of claim 35 further including the step of producing a digital signature for each data packet and including the digital signature in the data packet.

45. The method of claim 34 further including the step of producing a digital signature for the integrity block and including the signature in the block.

46. The method of claim 35 wherein the step of manipulating the selective data bytes includes concatenating the selected data bytes from a given data packet to produce the associated integrity check.

47. The method of claim 35 wherein the step of manipulating the selected data bytes includes encoding the selected bytes from a given data packet to produce the associated integrity check.

48. The method of claim 34 further including the step of including in a transmission extraneous packets that are associated with and do not pass one or more of the integrity checks.

49. The method of claim 34 wherein the step of encrypting the integrity checks further includes encrypting into the integrity block executable code that performs an integrity check process.

50. The method of claim 49 wherein the encrypting step further includes encrypting into the integrity block a digital signature associated with the code.

51. A data authentication system comprising:

A. an integrity block processor that receives one or more data packets and an associated integrity block, the integrity block processor manipulating the integrity block to produce one or more integrity checks that correspond to the data packets, and

B. an integrity check processor that uses the one or more integrity checks and selected data bytes from the one or more data packets to determine if any of the data packets have been altered.

52. The authentication system of claim 51 wherein the integrity block processor further produces from the integrity block information to determine which data bytes to select from the one or more data packets.

53. The authentication system of claim 51 wherein the integrity block processor decrypts the integrity block to produce the plurality of integrity checks.

54. The authentication system of claim 53 wherein the integrity block processor uses a shared secret key to decrypt the integrity block.

55. The authentication system of claim 53 wherein the integrity block processor decrypts the integrity block to provide to the integrity check processor executable code to use to manipulate the selected data bytes.

56. The authentication system of claim 53 wherein the integrity block processor further produces information to determine which data bytes to select from the one or more data packets by decrypting the integrity block

57. The authentication system of claim 53 wherein the integrity block processor uses a digital signature included in the integrity block to authenticate the integrity block.

58. The authentication system of claim 51 wherein the integrity check processor uses one or more digital signatures included in the one or more data packets to further authenticate the data packets.

59. A system for authenticating one or more data packets, the system comprising:

A. means for configuring at least one sending station with an authentication process adapted to produce an encrypted integrity block from one or more integrity checks associated with one or more data packets;

B. means for configuring at least one receiving station with an authentication process adapted to decrypt a received integrity block into one or more integrity checks and authenticate the associated one or more data packets using the one or more integrity checks.

60. The system of claim 59 wherein the means for configuring at least one sending station includes a computer readable medium containing executable program instructions.

61. The system of claim 59 wherein the means for configuring at least one receiving station includes a computer readable medium containing executable code.

62. The system of claim 59 further including means for configuring the sending station to transmit extraneous data packets that are associated with the integrity block but do not pass authentication.

63. A computer data signal embodied in a carrier wave and representing sequences of instructions for authenticating data packets, the instructions comprising instructions for:

configuring at least one sending station to produce an encrypted integrity block for one or more data packets; and

at the configured sending station selecting one or more data bytes from the one or more data packets and producing one or more integrity checks that are used to produce the encrypted integrity block.

64. The computer data signal of claim 63 wherein the selection of data bytes from a first data packet is random and the data bytes selected from remaining data packets are offset from the data bytes selected from first data packet.

65. The computer data signal of claim 63 wherein the integrity block is encrypted in accordance with a shared secret key.

66. The computer data signal of claim 63 wherein the one or more integrity checks are produced by concatenating selected data bytes from respective data packets.

67. The computer data signal of claim 63 wherein the one or more integrity checks are produced by encoding selected data bytes from respective data packets.

68. The data signal of claim 63 further comprising instructions for

configuring at least one receiving station to decrypt the encrypted integrity block to reproduce the one or more integrity checks; and

at the configured receiving station using the one or more integrity checks to authenticate the one or more data packets.

69. The computer data signal of claim 68 wherein the one or more integrity checks are associated with the appropriate one or more data packets prior to authentication.

70. The computer data signal of claim 63 further including configuring the sending station to transmit one or more extraneous data packets that are associated with the integrity block but do not pass authentication tests.

71. A data authentication system in which sequences of instructions for authenticating data packets are stored on a machine-readable medium, the instructions comprising instructions for:

configuring at least one sending station to produce an encrypted integrity block for one or more data packets; and

at the configured sending station selecting one or more data bytes from the one or more data packets and producing one or more integrity checks that are used to produce the encrypted integrity block.

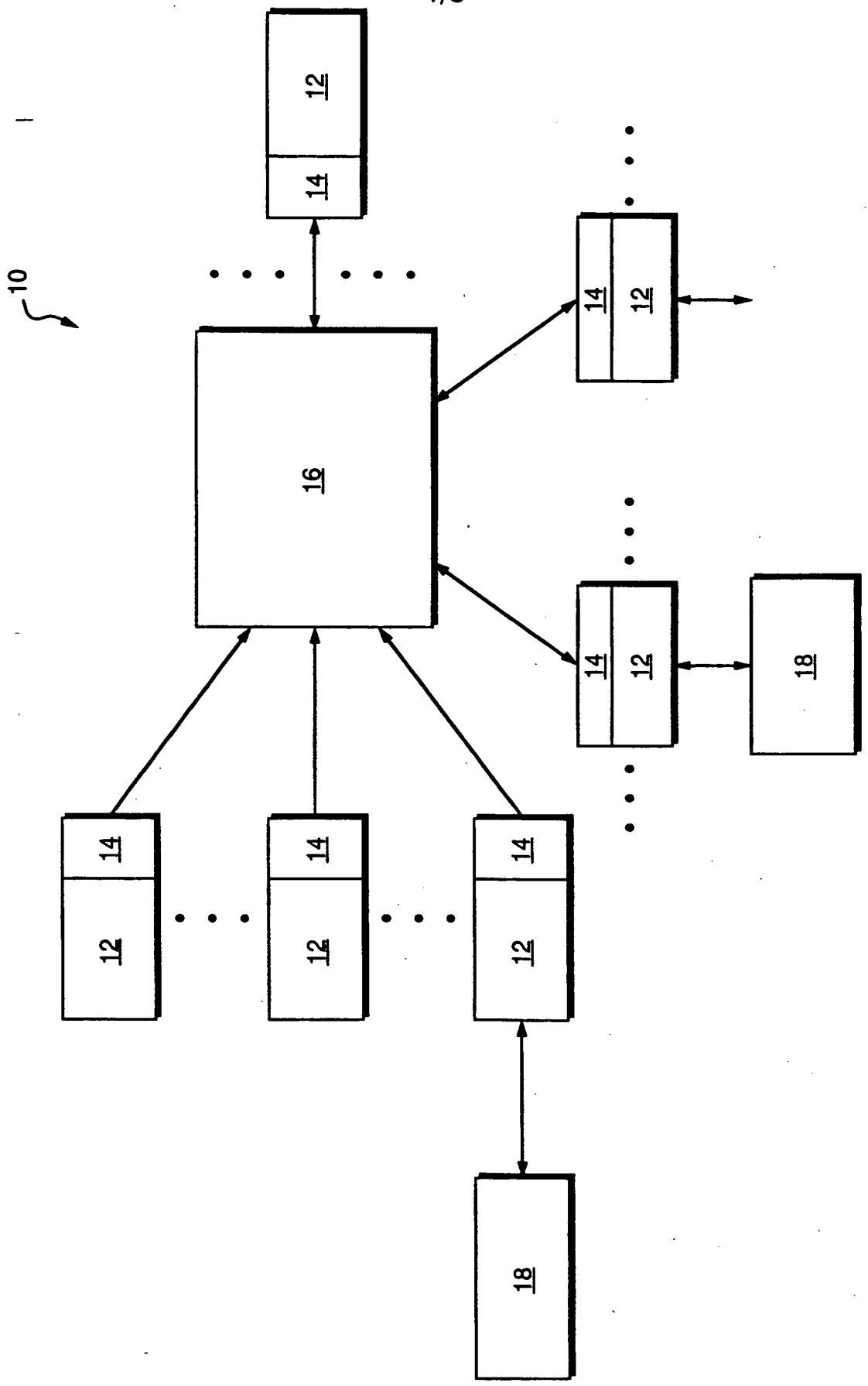


FIG. 1

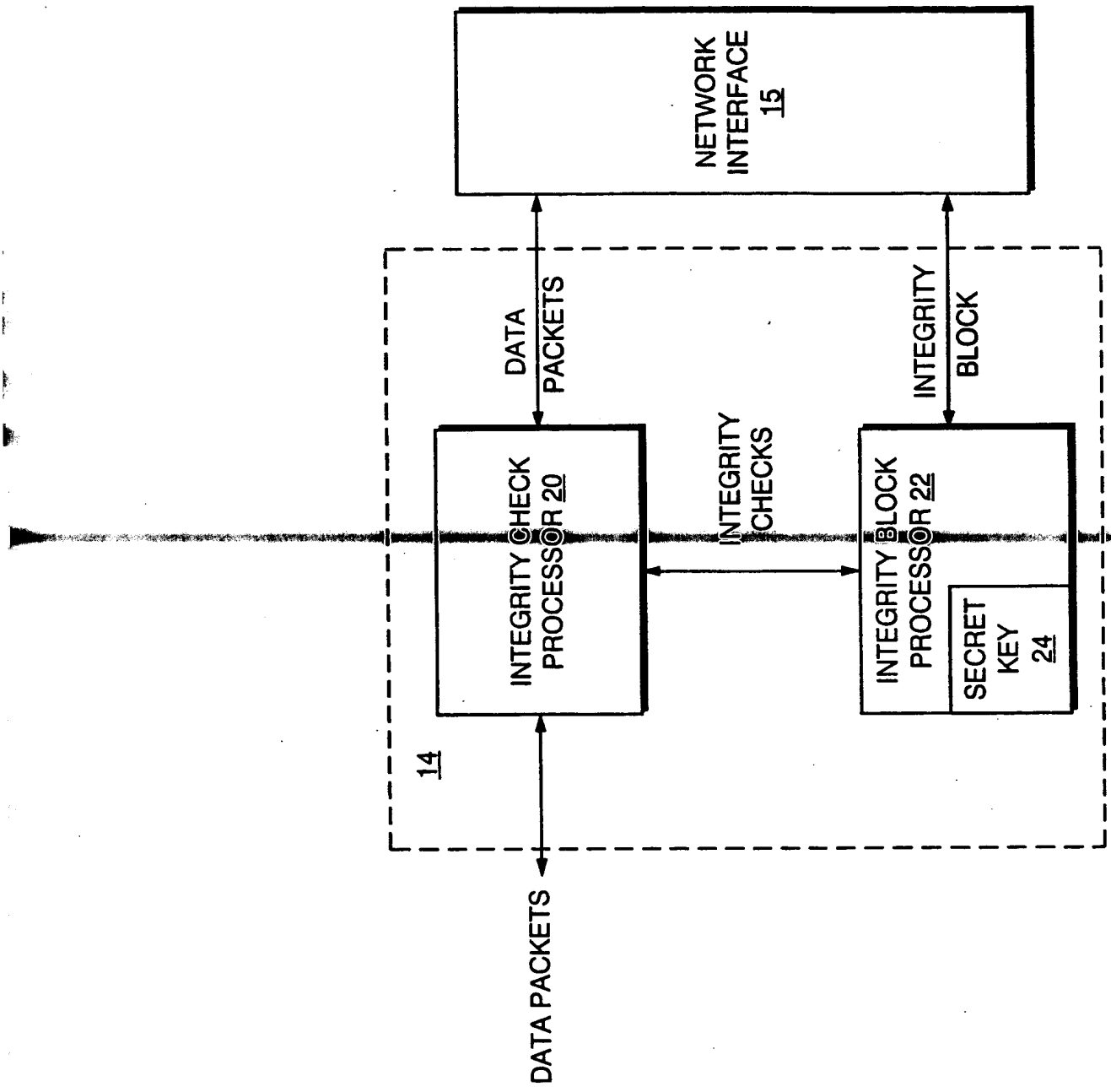


FIG. 2

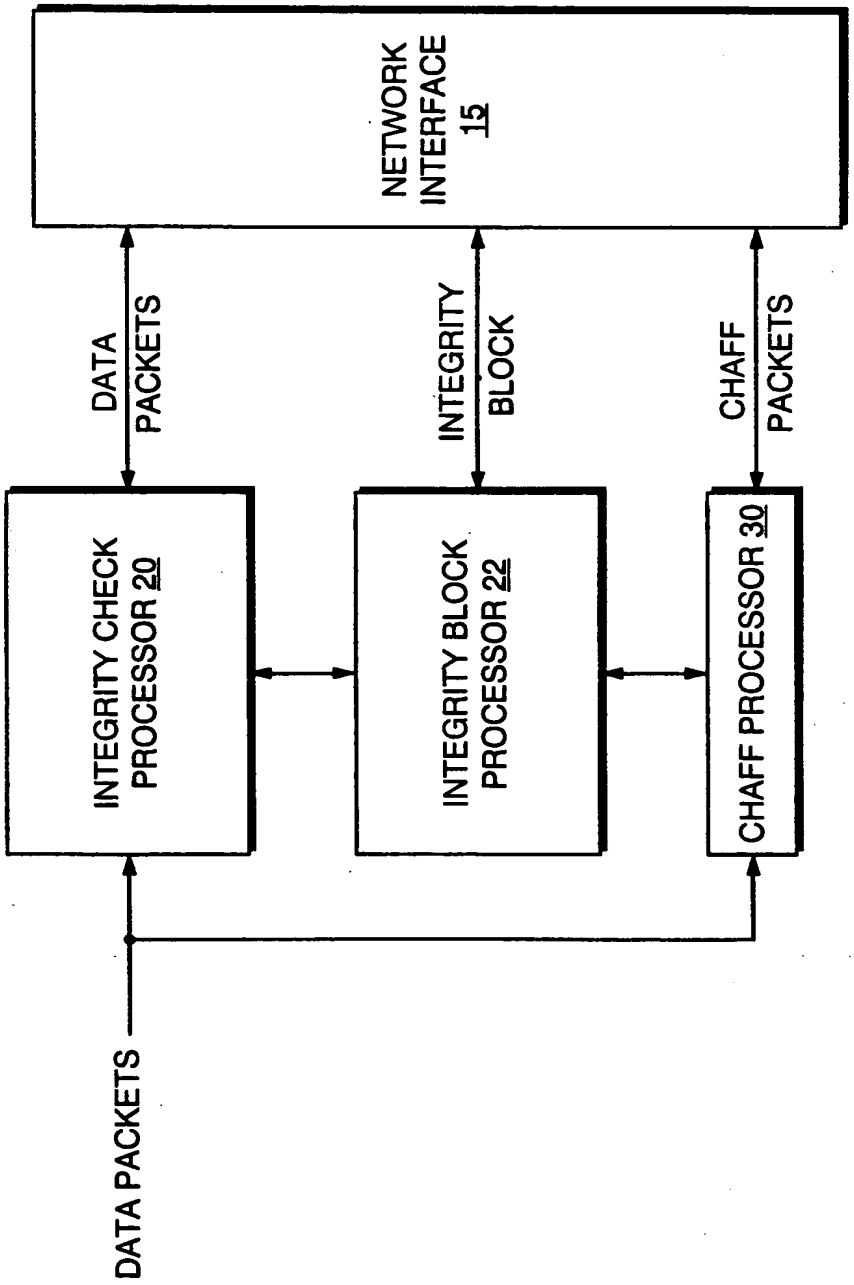


FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/03960

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) :H04L 9/00
US CL :713/151, 160, 161, 181
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 713/151, 160, 161, 181

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 5,440,633 A (AUGUSTINE et al) 08 AUGUST 1995, abstract	1, 2, 4-6, 11, 12, 16-20, 22-24, 28, 29, 33-37, 39-41, 46, 47, 51-54, 56, 59-61, 63, 65-69, 71 --- 3, 7-10, 13-15, 21, 25-27, 30-32, 38, 42-45, 48-50, 55, 57, 58, 62, 64, 70

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance, the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance, the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 17 APRIL 2000	Date of mailing of the international search report 12 MAY 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer GAIL HAYES James R. Matthews Telephone No. (703) 308-3900

Form PCT/ISA/210 (second sheet) (July 1998)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/03960

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,850,449 A (MCMANIS) 15 DECEMBER 1998, abstract, figures 6 and 7, column 4 lines 48-52	14, 15, 31, 32, 49, 50, 55
Y	US 5,349,642 A (KINGDON) 20 SEPTEMBER 1994, column 3 lines 65-68, column 4 lines 1-22, column 5 lines 10-21	9, 10, 27, 44, 45, 57, 58
Y	RIVEST, R. L. Chaffing and Winnowing: Confidentiality Without Encryption, MIT Lab for Computer Science, entire document, especially pp. 2	13, 30, 48, 62, 70
Y,P	US 5,948,119 A (BOCK et al) 07 SEPTEMBER 1999, column 24 lines 43-48	7, 8, 25, 26, 42, 43, 64
Y	US 5,266,942 A (STOLLER) 30 NOVEMBER 1993, column 4 lines 16-38	3, 21, 38, 64